Reinforcement Learning
Exercise 7

November 8, 2022

Introduction

In this exercise we will dive into model-based reinforcement learning and implement two algorithms. We use simulator rather than learning the dynamic model for this exercise.

Cross Entropy Method (CEM)

In this section, we will try to solve *CupCatch* environment from DeepMind Control Suite by planning using CEM. In this environment, an actuated planar receptacle can translate in the vertical plane in order to swing and catch a ball attached to its bottom. The catch task has a sparse reward: 1 when the ball is in the cup, 0 otherwise. In this exercise, we used a wrapper to repeat the same action 6 times, therefore, the maximum reward function for each timestep is 6.

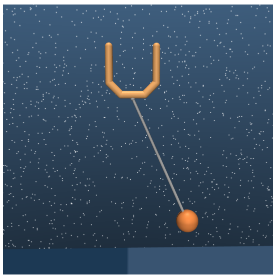

Figure 1: *Cup-Catch environment*

Task1 — 40 points Complete the code in `cem.py`.You need to complete planning part in the code. You can check *Joblib* for more information about paralleling. **Attach reward function plot in your report.**

The reference training plot is as Figure 2:

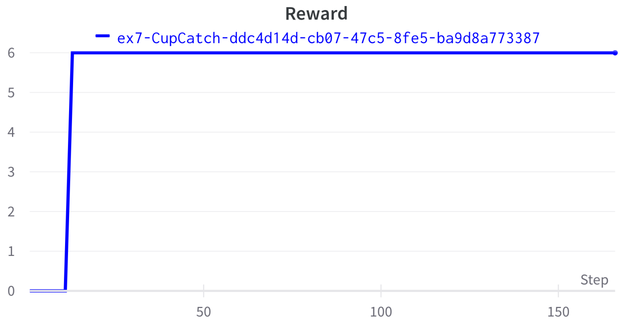

Figure 2: Reward function at each time step in CEM

Question 1.1 — 10 points Discuss the effect of changing *number of samples*. How can this affect the performance and running time?

Question 1.2 — 20 points Assume that the dynamic model is learned from data during training, compare this method (CEM with a learned dynamics model) with a model-free RL algorithm, such as DDPG. Please list two advantages and two disadvantages of this method. (Hint: think about a real-time application such as controlling humaniod robot)

AlphaZero

AlphaZero is an state-of-art algorithm which was able to taught itself how to master different games like chess and Go. In this section, we will use this algorithm to solve deep-sea environment form Behaviour Suite for Reinforcement Learning (bsuite). The environment targets the challenge of exploration and represents a N×N grid where the agent starts in the top left and has to reach a goal in the bottom right location. At each timestep, the agent moves one row down and can choose one out of two actions. The agent observes the current location and receives a small negative reward of -0.01/N for moving right and 0 reward for moving left. Additionally, the agent receives a reward of +1 for reaching the goal (treasure) and the episode ends after N timesteps. In this exercise, the number of rows and columns (N) is 10.

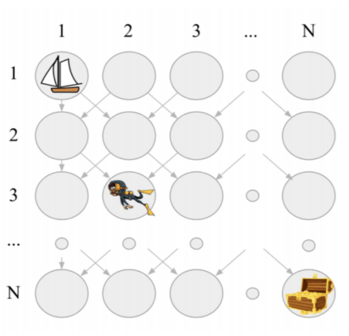

Figure 3: Deep-sea exploration: a simple example where deep exploration is critical.

Figure 3: *Deep-Sea environment*

Question 2.1 — 10 points Let's first think about the difficulty of the deep-sea task. 1) What is the probability of reaching the goal state (a function of N)? 2) If N is large, DQN (with the ϵ-greedy policy) usually fail to reach the goal state (in fact, N=10 is already challenging for DQN). In this case, which strategy will DQN converge to?

Question 2.2 — 10 points Describe different phases in MCTS.

Question 2.3 — 10 points Explain how using Actor Critic is beneficial in AlphaZero. Compare this to using Monte Carlo estimate of return function (Method you see on lecture slides).

Note: You can skim the paper (https://arxiv.org/abs/1911.08265). This paper is the MuZero, which is the AlphaZero with a learned dynamic model. We use this paper because it covers all required details and can be accessed by everyone. And you can use https://www.science.org/doi/10.1126/science.aar6404abstract if you can access.

Task 2.1 — 5 bonus points Implement PUCT search policy in Monte Carlo Tree Search (MCTS). You need to complete `puct` function in `MCTS` class, in `az.py`. Check the equation (2) in Appendix B

Task 2.2 — 25 bonus points Complete `mcts` function in `MCTS` class. Implement trajectory generation in MCTS. Complete `while node.children` loop, you need to select action, execute it and update information for the child node. In addition, you need to compute return function and update each node in the generated trajectory.

Task 2.3 — bonus 10 points Update Actor and Critic in `AZAgent` class. Hint: use TD target for Critic network. Check the equation (1) in section 3 but without the reward loss. (In our simple task, it should work even without the regularization.)

Task 2.4 — 10 bonus points One time use PUCT search policy and another use greedy policy (select action with highest value for each node). Compare the results and justifies them.

Attach training performance plot in your report (only for PUCT serach policy). The reference training plot is as Figure 4:

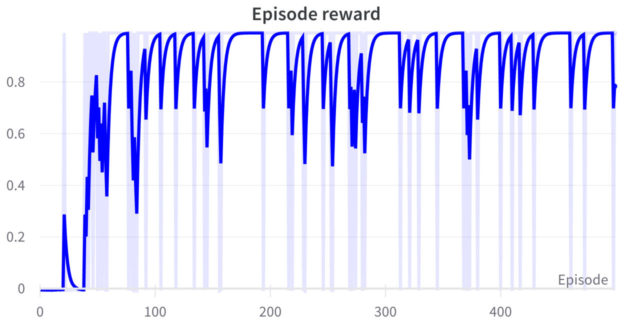

Figure 4: Training performance in AlphaZero

Submitting

The deadline to submit the solutions through MyCourses is on Monday, 21,11 at 23:55. Example solutions will be presented during exercise sessions that day.

Your submission should consist of (1) a **PDF report** containing **answers to the Questions** asked in these instructions and plots/model files/reported metrics **as required in each of the Tasks**, (2) **the code** with solutions used for the exercise. Please remember that not submitting a PDF report following the **Latex template** provided by us will lead to subtraction of points.

For more formatting guidelines and general tips please refer to the submission instructions file on MyCourses.

If you need help or clarification solving the exercises, you are welcome to join the exercise sessions.